IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A program storage device, readabl~~y~~e by a machine, tangibly embodying programming instructions to perform method steps for constructing a call graph as a representation of a program, the programming instructions comprising:

selecting a program P for constructing a call graph representation thereof;

wherein the program P contains zero or more fields F_F and one or more methods M_M,

wherein each method M_1 in M_M has a single body B;

wherein for each method M_2 in M_M, the call graph representation includes a corresponding node;

wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes;

~~determining~~identifying for each method M in M_M, a set of zero or more types S_M of objects ~~that may~~which occur therein ~~method M~~;

~~determining~~identifying for each field F in F_F; a set of zero or more types S_F of objects ~~that may be~~ stored therein ~~field F~~;

~~determining~~identifying one or more~~the~~ allocation sites inside the body B of each of method M;

determining ~~the~~a set of directly called methods M' inside the body B of each method M; and

determining ~~the~~a set of virtually called methods M" inside the body B of each method M.

2. (Currently Amended) The program storage device according to claim 1, further comprising the programming instructions of:

adding T to the types S_M for each allocation of type T that occurs in the method M.

3. (Currently Amended) The program storage device according to claim 2, further comprising the programming instructions of:

for each direct call to the methods M' in ~~a~~the body B of the method M performing the steps of:

adding any type that occurs in the types S_M and that is a subtype of ~~a~~the type of a parameter of the methods M' to types $S_{M'}$; and

adding any type that occurs in the types $S_{M'}$ and that is a subtype of ~~the~~a return type of the methods M' to the types S_M.

4. (Currently Amended) The program storage device according to claim 3, further comprising the programming instructions of:

for each virtual call to the methods M' in the body B of the method M:

using the types S_M , determine each of the methods M" that may be reached by ~~the~~a dynamic dispatch:

adding any type that occurs in the types S_M and that is a subtype of the type of a parameter of the methods M" to a set $S_{M''}$;

adding any type that occurs in the set $S_{M''}$ and that is a subtype of the return type of the methods M" to the types S_M.

5. (Currently Amended) The program storage device according to claim 4, further the programming instructions of:

for each field F read by the method M, add any type that occurs in the types S_F to the types S_M ; and

for each field F with the type T written by the method M, add any type that occurs in the types S_M and that is a subtype of the type T to the types S_F .

6. (Original) The program storage device according to claim 1, further comprising the programming instructions of:

using the call graph computed above in a compiler as a basis for performing optimizations such as inlining.

7. (Original) The program storage device according to claim 1, further comprising the programming instructions of:

using the call graph computed above in a reporting tool to report call graph information to a user.

8. (Currently Amended) A program storage device, readabl~~y~~e by a machine, tangibly embodying instructions to perform method steps for constructing a call graph as a representation of a program, the method comprising:

selecting a program P for constructing a call graph representation thereof;

wherein the program P contains zero or more fields F_F and one or more methods M_M,

wherein each method M_1 in M_M has a single body B;

wherein for each method M_2 in M_M, the call graph representation includes a corresponding node;

wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes;

~~determining~~identifying for each method M in M_M, a ~~only one~~ set of zero or more types S_M of objects ~~that may~~which occur therein~~ method M~~;

~~determining~~identifying for each field F in F_F, a ~~only one~~ set of zero or more types S_F of objects ~~that may be~~ stored therein~~ field F~~;

~~determining~~identifying one or more~~the~~ allocation sites inside the body B of each of method M;

determining a set of directly called methods M' inside the body B of each method M; and

determining a set of virtually called methods M" inside the body B of each method M.

9. (Currently Amended) The program storage device according to claim 8, further comprising~~ the steps of~~:

~~determining the set of directly called methods M' inside the body B of the method M; and~~

~~determining the set of virtually called methods M'' inside the body of method M.~~

adding T to the types S_M for each allocation of type T that occurs in the method M.

10. (Currently Amended) A method for constructing a call graph as a representation of a program, the method comprising:

selecting a program P for constructing a call graph representation thereof;

wherein the program P contains zero or more fields F_F and one or more methods M_M,

wherein each method M_1 in M_M has a single body B;

wherein for each method M_2 in M_M, the call graph representation includes a corresponding node;

wherein the call graph representation includes zero or more edges corresponding to connections between two or more of nodes;

~~determining~~identifying for each method M in M_M, a set of zero or more types S_M of objects ~~that may~~ occur therein~~ method M~~;

~~determining~~identifying for each field F in F_F, a set of zero or more types S_F of objects ~~that may be~~ stored therein~~ field F~~;

~~determining~~identifying one or more~~the~~ allocation sites inside the body B of each of method M;

determining ~~the~~a set of directly called methods M' inside the body B of each method M; and

determining ~~the~~a set of virtually called methods M'' inside the body B of each method M.

11. (Currently Amended) The method according to claim 10, further comprising:

adding T to the types S_M for each allocation of type T that occurs in the method M.

12. (Currently Amended) The method according to claim 11, further comprising:

for each direct call to the methods M' in athe body B of the method M performing the steps of:

adding any type that occurs in the types S_M and that is a subtype of ~~the~~a type of a parameter of the methods M' to types $S_{M'}$; and

adding any type that occurs in the types $S_{M'}$ and that is a subtype of ~~the~~a return type of the methods M' to the types S_M.

13. (Currently Amended) The method according to claim 12, further comprising:

for each virtual call to the methods M' in the body B of the method M:

using the types S_M , determine each of the methods M" that may be reached by ~~the~~a dynamic dispatch:

adding any type that occurs in the types S_M and that is a subtype of ~~the~~a type of a parameter of the methods M" to a set $S_{M''}$;

adding any type that occurs in the set $S_{M''}$ and that is a subtype of the return type of the methods M" to the types S_M.

14. (Currently Amended) The method according to claim 13, further comprising:

for each field F read by the method M, add any type that occurs in the types S_F to the types S_M ; and

for each field F with the type T written by the method M, add any type that occurs in the types S_M and that is a subtype of the type T to the types S_F .

15. (Original) The method according to claim 10, further comprising the step of:

using the call graph computed above in a compiler as a basis for performing optimizations such as inlining.

16. (Original) The method according to claim 10, further comprising the step of:

using the call graph computed above in a reporting tool to report call graph information to a user.

17. (Cancelled) A method for constructing a scalable call graph, the method comprising:

determining for each method M, only one set of types S_M of objects that may occur in method M; and

determining for each field F, only one set of types S_F of objects that may be stored in field F; and

determining the allocation sites inside the body of method M;

18. (Cancelled) The method to claim 17, further comprising the steps of:

determining the set of directly called methods M' inside the body of method M; and

determining the set of virtually called methods M" inside the body of method M.